

APPENDIX

Details of the Meta-analysis

The meta-analysis is an update of the previous 2002 meta-analysis that includes results from published trials of mammography screening for women aged 40 to 49 years that report reduction in breast cancer mortality. With the addition of only one new data point, the meta-analysis for the update was less extensive than the 2002 meta-analysis. We did not update the model for relative risk and length of follow-up (the two-level hierarchical model). We conducted similar updates for other age groups for context.

As with the original 2002 meta-analysis, we estimated the model by using a Bayesian data analytic framework, but this time using the BRugs package in R (22, 23). BRugs is an R interface to OpenBUGS, the successor to WinBUGS. The R code to create the dataset is below.

```
# R code to create dataset
study <- c('Age', 'CNBSS-1', 'HIP', 'Gothenburg', 'Stockholm', 'Malmo', 'Kopparberg', 'Ostergotland')
y.int <- c( 105, 105, 64, 34, 34, 53, 22, 31)
n.int <- c( 53884, 25214, 13740, 11724, 14303, 13568, 9582, 10285)
py.int <- c( 578390, 282606, 192360, NA, 203000, 184000, 124566, 172000)
y.cntl <- c( 251, 108, 82, 59, 13, 66, 16, 30)
n.cntl <- c( 106956, 25216, 13740, 14217, 8021, 12279, 5031, 10459)
py.cntl <- c(1149380, 282575, 192360, NA, 117000, 160000, 65403, 176000)
n <- 10000
rate.int <- n * y.int / n.int
rate.cntl <- n * y.cntl / n.cntl
rr <- rate.int / rate.cntl
rd <- rate.int - rate.cntl
nns <- 1 / ((y.cntl / n.cntl) - (y.int / n.int))
dataset <- data.frame(
  study,
  y.int, n.int, py.int, rate.int,
  y.cntl, n.cntl, py.cntl, rate.cntl,
  rr, rd, nns
)
```

APPENDIX

```
# Save dataset for BRugs to use
dataset.bugs <- cbind(y.int, n.int, y.cntl, n.cntl)
colnames(dataset.bugs) <- c("y.int", "n.int", "y.cntl", "n.cntl")
bugsData(data.frame(dataset.bugs), fileName="dataset.bugs", digits = 5)
constants <- cbind(nrow(dataset.bugs))
colnames(constants) <- c("n")
bugsData(data.frame(constants), fileName="constants.bugs", digits = 1)
```

The model assumes that the number of deaths from each study come from a binomial distribution with the probability parameter of α for the control group and $\alpha + \beta$ for the screening group. A random component, σz_i , is added to both probability parameters to allow for the random effect of the study i . Noninformative prior probability distributions were used.

```
# BUGS model
# This model is saved in a text file named "model.bugs"
model;
{
  for( i in 1 : n ) {
    z[i] ~ dnorm(0, 1)
    logit(p.int[i] ) <- alpha + beta + sigma * z[i]
    logit(p.cntl[i]) <- alpha      + sigma * z[i]
    y.int[i] ~ dbin(p.int[i] , n.int[i] )
    y.cntl[i] ~ dbin(p.cntl[i], n.cntl[i])
  }
  alpha ~ dnorm(-5.0, 1.0E-1)
  beta  ~ dnorm(0.0, 1.0E-1)
  sigma ~ dnorm(0.5, 1.0E-1) I(0, )
}
```

Four separate Markov chains with overdispersed initial values were used for estimation. A burn-in of 10,000 draws was used to initialize the chains and were checked for convergence.

```
# Check the model and load the dataset
modelCheck("model.bugs")
modelData("constants.bugs")
modelData("dataset.bugs")
```

APPENDIX

```
# Compile the model with 4 MCMC chains
modelCompile(numChains=4)
# Generate overdispersed initial values
modelGenInits()
# Keep MCMC samples of parameters alpha, beta, and sigma
samplesSet("alpha")
samplesSet("beta")
samplesSet("sigma")
# Thin samples so only 1000 draws are left
samplesSetThin(10000/(1000/getNumChains()))
# Generate 10,000 burn-in draws
modelUpdate(10000)
samplesHistory("*", thin=samplesGetThin())
```

The convergence of the parameter estimation was assessed and deemed adequate from the 10,000 burn-in draws. Next, we generated 100,000 draws from the four chains. These draws were thinned to yield a sample of 1,000 uncorrelated estimates from the posterior distributions.

```
# Clear samples from the previous burn-in
samplesClear("*")
# Keep MCMC samples of parameters alpha, beta, and sigma
samplesSet("alpha")
samplesSet("beta")
samplesSet("sigma")
# Thin samples so only 1000 draws are left
samplesSetThin(100000/(1000/getNumChains()))
modelUpdate(100000)
samplesHistory("*", thin=samplesGetThin())
# Check correlation of the thinned samples
for (i in 1:getNumChains()) {
  samplesAutoC("*", i, thin=samplesGetThin())
}
# Check the probability distribution of the parameters
samplesDensity("*", thin=samplesGetThin())
# Output sample estimates to an R object
brugs.nodes <- samplesHistory("*", thin=samplesGetThin(), plot=FALSE)
```

APPENDIX

After the model was estimated and the samples were thinned, sample rates per 10,000 women screened with mammography and control participants were calculated from the estimates of alpha and beta. Sample relative risk, risk difference, and number needed to invite to screening were calculated from the sample rates.

```
# Assign parameter samples to separate R vectors
alpha <- as.vector(brugs.nodes$alpha)
beta  <- as.vector(brugs.nodes$beta )
sigma <- as.vector(brugs.nodes$sigma)
# Rate calculations
# Note: this produces 1000 samples for each rate, RR, RD, and NNS
n <- 10000
rate1 <- n * exp(alpha+beta) / (1+exp(alpha+beta))
rate2 <- n * exp(alpha  ) / (1+exp(alpha  ))
rr <- rate1 / rate2
rd <- rate1 - rate2
nns <- n / (rate2 - rate1)
```

From the 1,000 thinned posterior samples, point estimates (mean) and 95% credible intervals (2.5 and 97.5 percentiles) for relative risk, risk difference, and number needed to invite to screening were calculated.

```
# Define R function; it will be used a number of times
brugs.nodessummary <- function(x, name) {
  Samples <- length(x)
  Mean <- mean(x)
  SD <- sd(x)
  MCMC.error <- sd(x) / sqrt(length(x))
  Median <- median(x)
  P.025 <- quantile(x, prob=c(0.025))
  P.975 <- quantile(x, prob=c(0.975))
  nodessummary <- data.frame(cbind(Samples, Mean, Median, P.025, P.975, SD, MCMC.error))
  rownames(nodessummary) <- name
  colnames(nodessummary) <- c("Samples", "Mean", "Median", "P.025", "P.975", "SD", "MCMC.error")
  data.frame(nodessummary)
```

APPENDIX

```
}  
# Call defined function brugs.nodesummary  
print(brugs.nodesummary(alpha , "alpha" ))  
print(brugs.nodesummary(beta , "beta" ))  
print(brugs.nodesummary(sigma , "sigma" ))  
print(brugs.nodesummary(rate1 , "rate1" ))  
print(brugs.nodesummary(rate2 , "rate2" ))  
print(brugs.nodesummary(rr , "rr" ))  
print(brugs.nodesummary(rd , "rd" ))  
print(brugs.nodesummary(nns , "nns" ))
```

The pooled number needed to invite to screening could be misleading if the baseline risk of mortality is appreciably varied between studies (67). One recommendation to accommodate this is to apply the pooled relative risk estimate to a range of control rates and then calculate number needed to invite to screening. The pooled rate of mortality among the control groups of our studies was estimated to be 35.5 deaths per 10,000 women (95% CrI 25.1 to 48.3). The range of mortality rates among the control groups was 16.2 to 59.7 per 10,000 women. Applying the pooled relative risk estimate of 0.85 to the high end of the mortality rate range (59.7) yields a number needed to invite to screening estimate of 1,116 per 10,000 women. Applying the pooled relative risk estimate of 0.85 to the low end of the mortality rate range (16.2) yields a number needed to invite to screening estimate of 4,115 per 10,000 women. This range 1,116 to 4,115 per 10,000 women is within the 95% CrI we report for number needed to invite to screening that we estimated from the posterior distributions of our mortality rate estimates. Alternatively, the bounds of our 95% CrI to number needed to invite to screening correspond to a range of control group mortality rates of 10.5 to 71.8 per 10,000 women, a range beyond that seen in the studies included in our analysis.